Introduction to GNU/Linux

Ivan Marković Matko Orsag Damjan Miklić
(Srećko Jurić-Kavelj)

University of Zagreb, Faculty of Electrical Engineering and Computing,
Departement of Control and Computer Engineering

2019

University of Zagreb
Faculty of Electrical Engineering
and Computing

The GNU/Linux operating system (History)

- Linux

```
Path: gmdzi!unido!fauern!ira.uka.de!sol.ctr.columbia.edu!zaphod.mps.ohio-state.edu!wupost!uunet!mcsun!ne
From: torva...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -
```

- GNU (GNU's Not Unix)

```
From CSvax:pur-ee:inuxc!ixn5c!ihnp4!houxm!mhuxi!eagle!mit-vax!mit-eddie!RMS@MIT-OZ
From: RMS%MIT-OZ@mit-eddie
Newsgroups: net.unix-wizards,net.usoft
Subject: new Unix implementation
Date: Tue, 27-Sep-83 12:35:59 EST
Organization: MIT AI Lab, Cambridge, MA

Free Unix!
```

The GNU/Linux operating system (Today)

Linux market share[1]

- 1.5% Personal computers
- 35% Servers
- 53% Smart devices
- 97% Supercomputers
- A comprehensive list of distributions `http://distrowatch.com/`
- Five most popular distributions according to DistroWatch: Manjaro, Mint, Ubuntu, elementary and Debian

[1] Source: Wikipedia

Installation alternatives

(a) Full Ubuntu installation on a dedicated partition

(b) Open Virtualization Archive Ubuntu 18.04 LTS

Casual Linux usage

- nothing to be afraid of :)
- Gnome and Unity "desktop managers"
- Dash menu (equivalent of Start in Windows)
- Web browsing: Firefox
- File browsing: Nautilus
- Text editor: Gedit
- Installing software: Ubuntu Software Center

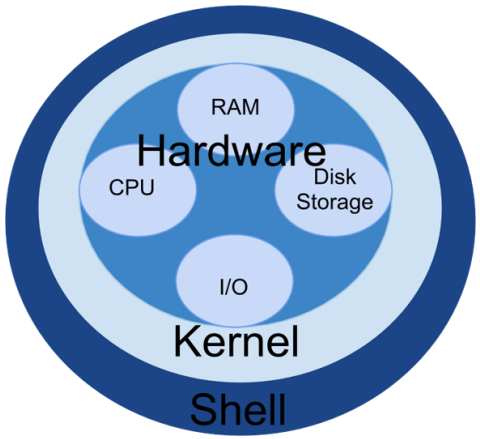
Exercise: Using Nautilus

Using the Nautilus file browser, create a folder named `pzros` in your home folder.

Power users use the shell (a.k.a command line)

Why use the shell, when we have icons, windows and mice?

- faster
- complex operations via chaining
- batch operations
- in some cases, it's *the only option*



We'll be using the bash shell.

A note on notation

- $ sign signifies a regular user prompt
- # sign signifies a superuser prompt
- Text in "monospace" font is to be entered literally, e.g.

```
user@host:~$ mkdir /tmp/test
```

- Pair of matching less/greater than signs (<>) denotes a "variable":

```
$ cd /home/<username>
```

- Pair of matching brackets denote an optional entry:

```
$ ls [-l]
```

Files and directories (1)

- Where are we?

```
user@host:~$ pwd
```

- What's inside?

```
user@host:~$ ls
```

- Additional options

```
user@host:~$ ls -la
```

- How to move/navigate?

```
user@host:~$ cd pzros
```

- Creating a directoy

```
user@host:~/pzros$ mkdir tmp
```

- Creating a file

```
user@host:~/pzros$ gedit README.pzros &
```

Files and directories (2)

(Very) useful tips

- Tab completion
- Command history

- How to list file contents?

```
user@host:~/pzros$ less README.pzros
```

- Copying a file

```
user@host:~/pzros$ cp README.pzros tmp/
```

- How to delete a file?

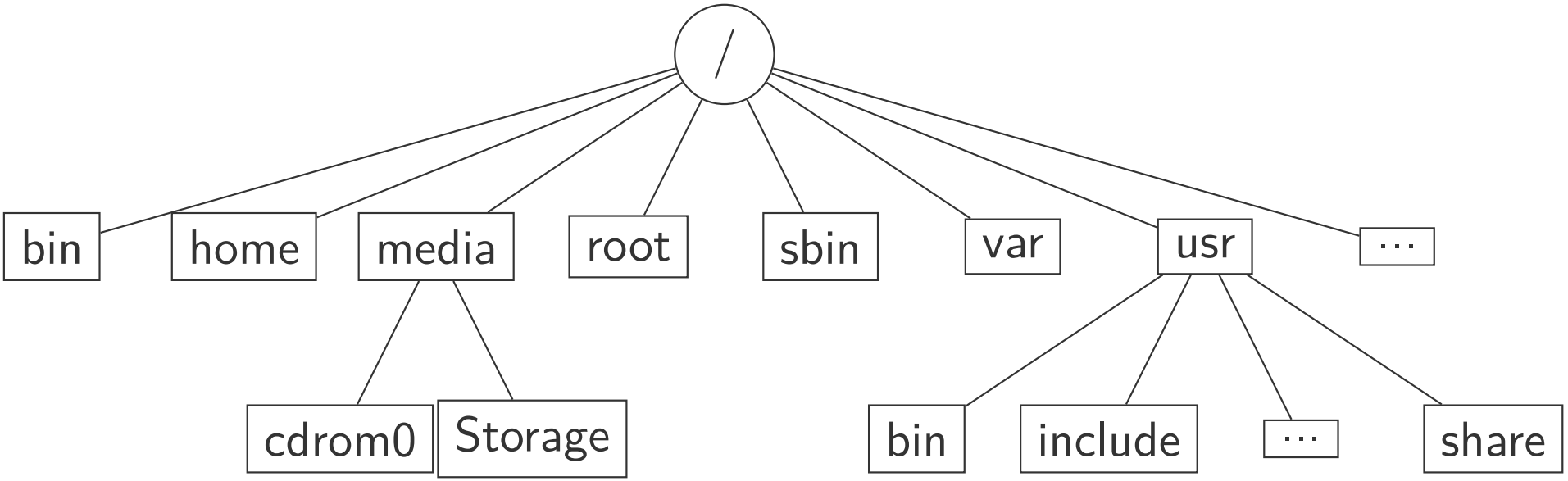
```
user@host:~/pzros$ rm tmp/README.pzros
```

Tip

Type *q* to get out of interactive programs like `less`.

Filesystem Hierarchy Standard

```
user@host:~/pzros$ ls -la /
```



`http://www.pathname.com/fhs/`

Exercise: Removable media

Insert a USB stick into your computer. List its contents in the terminal. Don't forget to eject the USB device before removing it!

Filesystem permissions

- List contents of the `root`[2] user home directory

```
$ ls /root
ls: cannot open directory /root: Permission denied
...
```

- Every file/directory has permissions set for three roles: owner, group and all others.

```
$ ls -ld /root
drwxr-xr-x 34 root root 4096 2012-09-21 17:00 /root
```

- Tools that manipulate ownership and permissions: `chown`, `chgrp` and `chmod`.

[2] This is where the term "to root" an Android device comes from

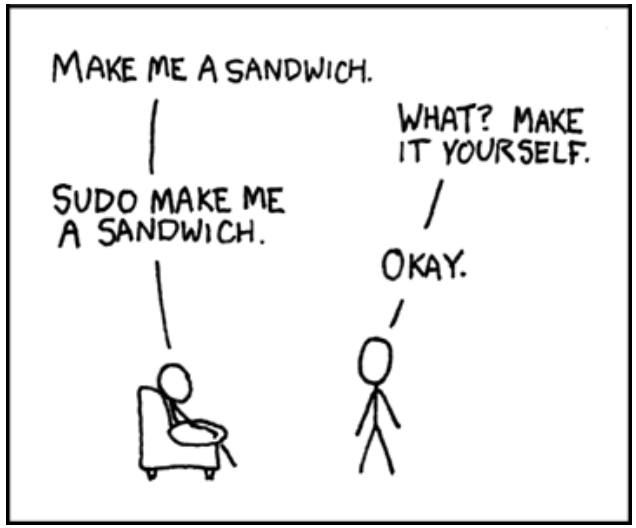
Superuser (root)

- Login disabled for root
- Multiple users can be designated as superusers using sudo[3]
- Executing a command with superuser privileges

```
user@host:~$ sudo ls /root
```

- Gaining a root shell

```
user@host:~$ sudo -i
root@host:~# ...
```



[3] Through the "User Accunts" GUI tool, or by editing (/etc/sudoers)

Command chaining and I/O redirection

- Pass a long output of a command to `less` using pipe |

```
$ ls -la /usr/lib | less
```

- Redirect the output of a command to a file

```
$ echo "Robots rule" > out.txt
$ cat out.txt
$ echo "Neces razbojnice\!" > out.txt
$ ls -la /usr/lib >> out.txt
```

`stdin`, `stdout` (1) and `stderr` (2) streams.

Getting help

- Almost all programs/commands have built-in help

```
$ ls --help
```

- For in–depth help use info...

```
$ info mkdir
```

- ...or man pages.

```
$ man mkdir
```

Exercise: `mkdir` and `ls` options

1. Create a nested folder `tmp2/tmp3/tmp4` inside of your `pzros` folder, using only one command (hint: check the options of the 'mkdir' command).
2. Inside of the `tmp4` folder, create a text file containing a list of all files from the `/usr/include` folder, ordered by timestamp.

Searching for files

- Searching for files

```
$ find /home -name README.txt
```

- Indexed searching with *locate*

```
$ locate stdio.h
```

- Find out which executable is run when a command is invoked

```
$ which info
/usr/bin/info
```

Searching inside files

- The *grep* command

```
$ grep <PATTERN> [FILE]
```

- Search for a string in a file

```
$ grep printf /usr/include/stdio.h
$ grep printf /usr/include/*.h
```

- grep can be very useful when combined with other commands using |

```
$ ls /usr/lib | grep python
```

Exercises

Exercise: `locate`

How many stdio.h files are there on your system. (Hint: use the locate command with an additional argument)

Exercise: `grep`

List all directories within the `/usr/lib` directory that have `rwxr-xr-x` permissions. Store this list to a text file. (Hint: Pass the output of `ls` to `grep`).

Advanced Packaging Tool (apt)

- Library with various front-ends:
  - Ubuntu Software Center
  - `apt-get`
- Software repositories in `/etc/apt/sources.list.d`
- Packages are signed and have extensive metadata
- Information is cached
- Updating the cache

```
$ sudo apt-get update
```

- Searching the cache

```
$ apt-cache search terminal | grep -i drop-down
```

- Installing software

```
$ apt-get install guake
$ sudo !!
```

Homework

Preparation

Install the ROS Melodic Morenia distribution (Desktop-Full variant), by following the official instructions from the ROS Wiki. (If ROS is already installed on your machine, you do not need to reinstall, simply skip this assignment.).

Assignments

1. What is the location of the program that gets executed with the `roscore` command?
2. Determine the sizes of the largest and smallest files located in the same folder as `roscore`. Report the sizes in human-readable format (K, MB etc.). (Hint: use options of the `ls` command to sort the files by size.)

Homework

Assignments (continued)

3. Find all files inside `/opt/ros/melodic` which are executable and whose name begins with `rqt_`. Store the output in a file named `3.txt` (Hint: use the `find` command with additional options).
4. In order to determine which Python programs depend on the tf coordinate transform library, find all files inside `/opt/ros/melodic` that contain the string `import tf` and count them. Store the output in a file named `4.txt` (Hint: use the `grep` command with additional option for recursive searching and the `wc` command.).

Summary

- Navigating the filesystem
- Creating files and folders
- Searching for things
- Process management
- Installing packages

Useful links

- Basic Use of Linux Operating System at UniZg FER
- Linux tutorial at University of Surrey
- Introdction to Linux edX online course
- Linux System Administration at Rutgers University
- Rute User's Tutorial and Exposition - free Linux usage and administration hanbook (somewhat outdated)
- Ubuntu documentation (official and community wiki)
- Ask Ubuntu (part of StackExchange network)

dpkg

- List installed packages

```
$ dpkg-query -l
```

- Find a package owning file

```
$ dpkg-query -S stdio.h
```

- List all files from a package

```
$ dpkg-query -L guake
```

Environment variables

Variables that affect the behavior of your system and processes.

- List all environment variables

```
$ env
```

- See the contents of a specific variable

```
$ echo $SHELL
```

- Set an environment variable

```
$ [export] PATH=$HOME/scripts:$PATH
```

Environment variables required in every shell session should be exported in `/.bashrc`

Process management (1)

- Suspending an app with `Ctrl+Z`

```
$ gedit
```

- Moving an app to background

```
$ ^Z
$ jobs
$ bg %1
```

- Killing an app

```
$ kill [-9] %1
```

- Starting an app in background or `Ctrl+C` while the app is in the foreground.

```
$ gedit &
```

- Bringing an app to foreground

```
$ fg %1
```

Process management (2)

- List all processes

```
$ ps [aux]
```

- Monitor processes

```
$ top
```

- Kill a process by PID

```
$ kill -9 <PID>
```

- A fancy kill using `pgrep` and backquote expansion

```
$ kill -9 `pgrep gedit`
```